NTNU – Trondheim
Norwegian University of
Science and Technology

TLS and the future of authentication

Dag Erik Vikan

Master of Science in Informatics
Submission date: Januar 2015
Supervisor: Tor Stålhane, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Contents

Chapter 1

Introduction

The purpose of this thesis is to summarize the latest research on authentication problems in the web public key infrastructure and to find and compare proposed improvements of doing authenticity in TLS [1].

The current system of providing authenticity in TLS (transport layer security) has some problems. 1832 root and intermediate certificate authorities (CA) from 683 organizations in 57 countries can currently sign X.509 certificates for any domain and be trusted by popular browsers [1].

CAs can be hacked by outsiders or subverted by insiders. They can make honest mistakes or behave at above acceptable risks or be coerced through political means.

With an illegitimate certificate (but CA-signed), an attacker may successfully man-in-the-middle (MITM) their victims.

The best known recent failure was the Diginotar CA compromise. In the attack, fake certificates for google.com and others where used and deployed in MITM attacks on Iranian users [2].

In 2013 the CA Turktrust accidentally issued two certificates with X509v3 Basic Constraints set to TRUE which marks the certificate an intermediate CA [3]. One of the customers used it on their local network.

In 2012 the CA Trustwave issued an intermediate CA certificate to a customer [4].

In 2011, a hacker originating from Iranian IP addresses compromised the CA Comodo. Resulting in 9 illegitimate certificates for well-known web sites [5]. Bogus certificates were issued for webmail systems, which were in turn

[1] We will refer to either "SSL" or "TLS" as "TLS".

used to intercept web traffic in Iran.

There are numerous other examples of CA breaches that shows this is a real problem with serious consequences. In addition to the structural problems, there are other issues. E.g. proof of ownership for DV (domain validated) certificates is typically being able to receive an email on the email address listed in WHOIS data. Inside is an HTTP URL with a secret token that must be clicked. Proof of ownership is complete after clicking the URL. The WHOIS protocol itself lacks security mechanisms. And the email systems itself can be insecure.

There is also a usability issue at play: allowing users to click through certificate validation errors defeats the purpose of encryption. Users click through these warnings. The recent HTTP Strict Transport Security (HSTS) [31] standard instructs compliant browsers to not allow users to click through these warnings, thus hard-failing.

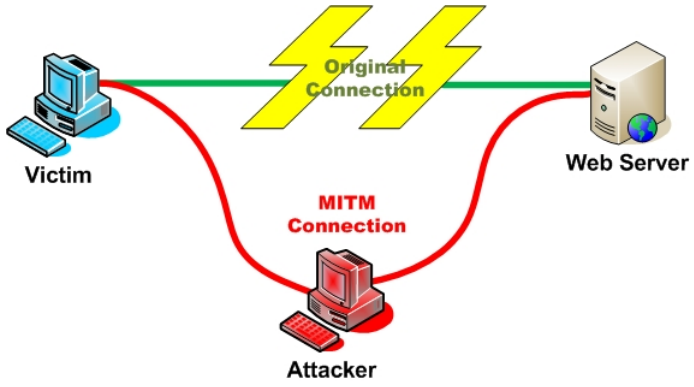

Figure 1.1: The classic man-in-the-middle attack.

1.1 Research questions

The research questions are:

1. Which alternative methods of providing authenticity in Transport Layer Security (TLS) exists?

2. What are the strengths and weaknesses of the alternative methods?

1.2 Contribution

This thesis gives a comprehensive overview of the latest research on authentication in TLS.

1.3 Motivation

Security is the topic I am most fascinated by and find most interesting. During my studies I learned about TLS and its problems. My motivation for writing this thesis is that I want to learn as much as I can about TLS.

1.4 Methodology

To produce useful answers to the research questions I will conduct a systematic literature review. The methodology requires considerably more effort than a traditional literature review. Prior to undertaking the review I tried to find similar efforts in the literature, since I do not want to duplicate recent effort. I endeavored an unsystematic approach on the Google search engine. I used the search string "review TLS SSL". I was unable to find anything similar.

Other researchers should be able to follow the same procedures and get the same results.

The systematic literature review (SLR) method is concisely explained by Kitchenham [6]:

- SLR defines a review protocol specifying research questions and search venues
- SLR involves explicitly documenting the search strategy and review procedure, so the reader can assess its rigor and completeness and a replication of the review is possible in the future
- SLR involves having inclusion/exclusion criteria that aid in identifying studies relevant to the purpose of the review
- SLR involves evaluating the quality of the reported studies

The three main phases are: plan the review, implement the review and report the review.

A predefined search strategy is needed. Explicitly describing the search strategy reduces the chances of bias. It also makes it repeatable. The aim is to detect as much of the relevant literature as possible. I cannot read all texts from the search results since it would be too time-consuming. When conducting the search I will read the title on each hit. If the title relates to authentication in TLS, I add it to the reading list

In an attempt to perform an exhaustive search I have identified three electronic sources of relevance:

1. IEEExplore
2. ACM Digital library
3. Google scholar

I chose these because they are the best known computer science libraries.

The search strings I will use are: "TLS", "SSL", "HTTPS" and "PKI".

I must decide which publications to include. This selection is governed by inclusion and exclusion criteria. I want to include anything that contributes to answering the research questions. The inclusion criteria are:

- Any study that argues for an improvement in the authentication in TLS.
- Published earliest in 2010

The exclusion criteria are:

- Studies concerned with the mathematical properties of crypto systems.
- Non-English studies.

Chapter 2

Theory

Few people truly understand computer security. To minimize the chance of accidentally joining this crowd we need to define some terms.

2.1 What is trust?

Figure 2.1: A group of people being trusted

Trust is a psychological state comprising expectancy: the trustor expects a specific behavior of the trustee such as providing valid information or effectively performing cooperative actions. Just like java, the word "trust" is also overloaded. I use the following definition: an entity can be said to "trust" a

second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects. Or equivalently, an entity estimates misbehavior from second entity to be unlikely.

Using this definition will make it absolutely clear what is meant by trusting a CA. To trust a CA means that the trustor estimates misbehavior to be unlikely.

2.2 Probability-Theoretic Observation

Before even beginning to investigate matters, an observation regarding chance should be made. Assume a list of n trusted root CAs. Each CA is compromised with a probability of $0 \leq p_i \leq 1$. The probability that no CA is compromised is the product of the probabilities that all CAs are not compromised. This can be expressed as $1 - (1 - p_i)^n$.

It is difficult to estimate the value of the expression, but assume that the probability that one CA is compromised is 0.01. For $n = 100$, the probability that at least one CA is compromised is 0.64. It can be seen that after considering a little over 200 CAs, the probability is 90% and it quickly approaches 100%. But even with a modest probability of 0.001, which is probably too low, there is 84% chance that at least one CA is compromised.

Figure 2.2 plots the probabilities that at least one CA is compromised with estimates of 0.01 and 0.001.

2.3 Threat model

We assume an active MITM attacker who may control network nodes. To focus effort on the authentication component only we will assume that the implementation of TLS have no errors and that the network end nodes do not have malware. I make these assumptions because I do not want to discuss implementation problems. I assume that the end nodes are malware-free because if not, there is no need for a secure protocol because the malware can bypass it anyway.

The consequences of these assumptions are that I need only concern myself with the main topic of discussion, namely the authentication component in TLS.

In short, an adversary can generate, modify, delete or delay traffic.

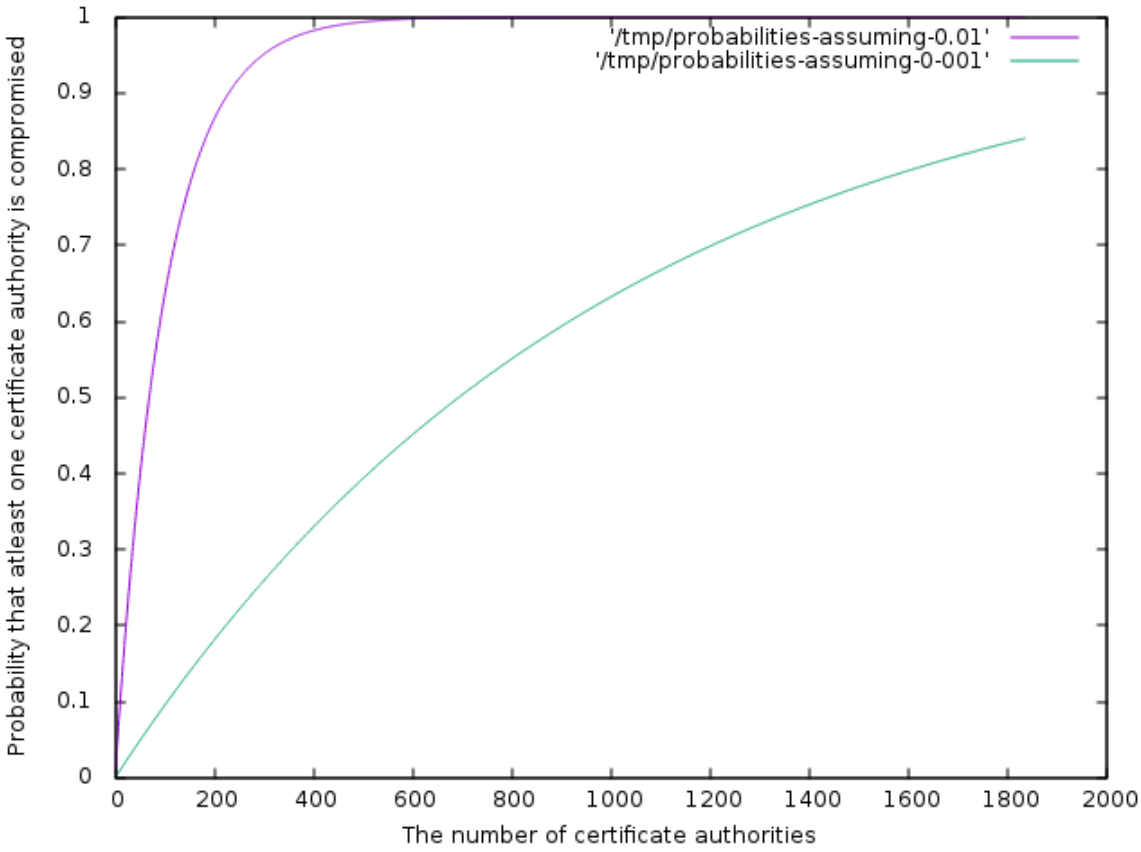

Figure 2.2: A plot of the probability that at least one CA is compromised when considering a compromise chances of 0.01 and 0.001.

2.4 SSL and TLS

TLS is a protocol for providing secure communication over the internet. Its predecessor is SSL which was developed by Netscape in 1994. The SSL specifications are from 1994, 1995 and 1996. TLS version 1.0 was standardized as RFC 2246 in 1999 by Internet Engineering Task Force (IETF). TLS 1.1 came in 2006 and TLS 1.2 came in 2008. The next version is currently in development. The protocol encapsulates the entire application layer packet [7].

A secure protocol should provide three things:

- Confidentiality (observer cannot read plain text)

- Integrity (no modification of data without detection)
- Authenticity (making sure you really are talking to correct entity)

In TLS, confidentiality and integrity is achieved using classic symmetric cryptography primitives. Keys are shared using asymmetric cryptography. The security of asymmetric cryptography assume certain mathematical conjectures being true. E.g. existence of one-way functions. The conjectures are not proven, but we think they are true. In the case if RSA, we assume that it is unfeasible to factor a large prime product. In mathematical terms a one-way trapdoor function is constructed. In asymmetric cryptography, a public/private key pair is generated. Anyone can use the public key to encrypt messages. Only the holder of the private key is able to decrypt.

The critical asset which the attacker is interested in, is the plaintext TCP packet. More specifically for clarity and concreteness, we can say the critical asset is the password submitted in an HTML form.

In computer science, authentication is the act of confirming the truth of a claim of identity. Alice typing away at her keyboard occasionally authenticates herself towards various computer systems. To authenticate herself she must first put forward a claim of her identity. Then she needs to prove that the identity claim is true. The computer as a proxy conducts the communication. The most common form of authentication is passwords. In this case, authenticity is implied but not guaranteed.

More generally, a person is not really proving that his personhood is identical to the claimed identity. In essence the authentication process is

1. Assumption: Only the person Alice knows the password for Alice's user.
2. A person claims to be Alice and knows her password.
3. Conclusion: This person must be Alice.

Traditionally, an identity claim is proved by providing a password. This is something that the user knows. This piece of data does not exists in the real world. It exists only inside the mind of a person.

The other two methods for proving identity claims are ownership and inherence. A person can own a physical piece of property such as a cellphone or hardware-token. A person is or does something. Examples are fingerprint, retinal pattern and voice.

In essence, network nodes want to prove to other network nodes that they are who they claim they are. When an HTTP client connects to Gmail, the Gmail network node authenticates itself by providing an X.509 certificate. Embedded in this certificate is a cryptographic signature. The producer of the signature is a trusted third party. They are called certification authorities (CAs). Their job is to certify cryptographic identities. In practice, a domain name owner needs simply to prove ownership of a domain. After the CA is done certifying, an X.509 certificate is provided. A small fee ranging between 0 to 100 USD is paid.

The key observation here is that a third party must be trusted by its clients. Embedded in browsers and the operating system (OS) lies a list of trusted CAs.

The authentication problem is how to communicate securely with an entity you never have communicated with before. It may be a problem with no solution [8]. It appears to be unsolvable because each time someone tries to design a cryptographic protocol, they assume a pre-existing shared secret. In the current CA-system, an operating system has pre-installed a list of CAs with their public keys. Where did the user get his copy of the operating system? Most often it is pre-installed. Following the trust chain all the way backwards you meet the hardware. We assume the fresh OS install and hardware do not contain malware.

2.5 Public Key Infrastructure

X.509 is an ITU-T standard for a public key infrastructure (PKI) and has been adopted by the Internet Engineering Task Force (IETF) as the PKI for several IETF protocols [19].

The main objective of a public key infrastructure (PKI) is to securely distribute public keys. A PKI system consist of the following components:

- Certificate: binding between public key and identity of entity
- End entity: users, devices, systems
- Certificate Authority: the issuer of certificates and revocations
- Revocation service: provides information related to revocation

The CA system is concisely described by [18]:

> The CA system exists to authenticate one party to another in a public-key infrastructure (PKI). Although client software ultimately carries out the authentication, CAs issue the digital certificates that make the authentication possible. Software vendors, at their discretion, build into their products a list of "root" CAs that are trusted to perform authentication on behalf of users.

2.5.1 Revocation

There are two standards for certificate revocation. A *Certificate Revocation List* (CRL) involves each CA periodically issuing a signed datastructure consisting of a list of serials. Revoked certificates are identified by their serials which are unique in the CA. The *Online Certificate Status Protocol* (OCSP) allows for clients to obtain the revocation status of a certificate [9]. To improve privacy and performance problems, *OCSP stapling* is a TLS extension which allows servers to piggyback an OCSP response onto the TLS handshake.

Browsers soft-fail when OCSP lockups time out. An attacker need simply drop the OCSP lookup. Soft-failing means they continue as if nothing happened. They soft-fail because they do not want to inconvenience their users just because the OCSP responder is down.

Revocation does not work well because browsers soft-fail and because there is a delay in propagating revocation information to each system. Browsers instead quickly patch their system. If the illegitimate certificates are missing revocation information it becomes unrevokable. This was a mistake in the design. Instead a cryptographic hash over the entire certificate is ideal.

OCSP remains a valid defense against situations where the attacker is not a MITM (code-signing or certificates issued in error).

2.6 SSL stripping

Prior to creating a secure connection with TLS there is often a bridge from secure to non-secure. This takes form in HTTPS URLs or a HTTP 302 redirect. This opens up for an attack where the attacker tricks the browser into never setting up a TLS connection. E.g. all HTTPS links can be converted to HTTP. For continued browsing to function, the attacker must remember which links it has stripped, and proxy out those when the victim requests

those resources. If the server delivers same content on non-TLS no such proxying is needed. The only difference a victim will see is that the HTTPS is missing from the URL in the case of a DV certificate. The green URL bar will be missing in the case of EV (extended validation) certificates.

As shown by Moxie Marlinspike with his tool SSL strip[30], this attack has a very high success rate. Field testing shows it has a 100% [1] success rate.

Intuitively, a server needs to inform clients that it communicates only via TLS. A TLS-only policy on the entire internet would remedy the situation. Directly typing the HTTPS URL in the browser URL bar or a browser bookmark also reflects the attack.

HSTS (HTTP Strict Transport Security) requires TLS-only with a server initiated pin in the form of an HTTP header [31]. Similar in spirit, EFF's *HTTPS Everywhere* [2] is a client enforced Firefox addon with a preloaded list of TLS-only pins [32].

[1] DEFCON 17: More Tricks For Defeating SSL: `https://www.youtube.com/watch?v=ibF36Yyeehw`

[2] `https://www.eff.org/https-everywhere`

Chapter 3

Literature review

3.1 iexplore

Searching for "TLS" gives 529 results. I have inspected 25 and have chosen the following:

`Detecting and defeating advanced man-in-the-middle attacks against TLS`
`A Notary Extension for the Online Certificate Status Protocol`
`Certification Authorities Under Attack: A Plea for Certificate Legitimation`

Searching for "SSL" gives 426 results. I have inspected 17 and have chosen the following:

`The potential of an Individualized Set of Trusted CAs: Defending against CA Failures in the Web PKI`
`SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements`
`Simple and Lightweight HTTPS Enforcement to Protect against SSL Striping Attack`

Searching for "PKI" gives 347 results. I have inspected 22 and have chosen:

`The X.509 trust model needs a technical and legal expert`
`Not Reinventing PKI until We Have Something Better`
`Trust Darknet: Control and Compromise in the Internet's Certificate Authority Model`

3.2 ACM digital library

Searching for "TLS" gives 778 results. I have inspected 22 and have chosen:

`Global authentication in an untrustworthy world`
`No attack necessary: the surprising dynamics of SSL trust relationships`
`Analysis of the HTTPS certificate ecosystem`
`Rethinking SSL development in an appified world`
`Security Collapse in the HTTPS Market`
`Certificate transparency`

Searching for "SSL" gives 1319 results. Too many results. Sorted by date and only considered the first 500. I have inspected 6 and have chosen:

`Accountable key infrastructure (AKI): a proposal for a public-key validation infrastructure`

Searching for "PKI" gives 772 results. Too many results. Sorted by date and only considered the first 500. I have inspected 6 have chosen none.

3.3 Google scholar

Searching for "TLS" gives 783 results. Too many hits. Sorted by relevance and only considered the first 500. I have inspected 24 and have chosen:

`Public Key Pinning for TLS Using a Trust on First Use Model`
`PoliCert: Secure and Flexible TLS Certificate Management`

Searching for "SSL" gives 527 results. I have inspected 10 and have chosen:

`Certified Lies: Detecting and Defeating Government Interception Attacks against SSL (Short Paper)`

Searching for "PKI" gives 391 results. I have inspected 14 and have chosen none. In total 19 publications are selected for inclusion. Many others were discarded after inspection when yielding no useful data.

Chapter 4

Results

Five papers were published in 2014, ten in 2013, three in 2012 and one in 2011. And here is the frequency of terms relating to authentication:

Term	Frequency
DANE	13
Perspectives	13
Convergence	12
Sovereign Keys	10
Certificate Transparency	10
Certificate Patrol	9
HPKP	8
TACK	7
EFF SSL Observatory	5
MonkeySphere	4
MECAI	2
AKI	2
Crossbear	2
DoubleCheck	1
S-links	1
DNSChain	1
HSTS Cert pinning	1
PoliCert	1
DetecTor Project	1
ICSI Certificate Notary	1

4.1 Summaries

In the following is a short summary of each paper.

[10] asserts that most existing attempts at improving authentication is done by maintaining the current PKI model and by using certificate pinning. It also stresses that with the DANE (see section 5.1) approach, client must fail when the DANE lookup fails.

[11] suggests an extension to OCSP providing a notary service similar to Perspectives/Convergence (see section 5.3). Notaries should be run by CAs and not whoever cares to set up one. They highlight the problem with fake certificates with serials omitted. These are indeed unrevokable. Also, it is likely that the vast majority of users would not change the default list of notaries.

[12] introduces the notion of "certificate legitimation" and asserts high probability of repeated CA security problems. Using the X.509 name constraints extension is not going to cut it because an attacker can control the content of fake certificates. Slimming down the trust CAs list can be a viable alternative.

[13] created a tool which analyses browser history and finds only CAs needed. The list of trusted CAs could be reduced by 90% for one user.

[14] surveys and categorizes prominent security issues with HTTPS and raises concerns with the TLS protocol itself. Protocol attacks are found even after 16 years. It may be too complex.

[15] proposes a better defense against SSL Stripping attacks (see section 2.6).

[16] introduces a new role of technical and legal expert into the X.509 trust model to help the relying party (RP) by reading and analyzing the set of technical and legal documents provided by each CA.

[17] mentions that the PKI community never succeeded in solving the core problem of automatic key enrollment. The author also explains that DANE is unlikely to replace the current X.509-based PKI deployments because it would merely replace the too-many unscoped trust points problem with a potentially much worse too-many-registrars problem.

[18] explains that studies have repeatedly shown that users do not understand the concept of trusted CAs, or even care about TLS error dialogs.

The RFC 5280 [19] describes the "name constraints" extension which is to be used for limiting what entities and their name they can certify [20]. The usage of this extension has been negligible probably due to the desires

of CAs to certify as much as possible. The paper suggest a fundamental shift allowing clients to make their own decisions about trust. They suggest a "Policy Engine" utilizing many sources of data to predict impact of trust decisions and to actually make them.

[21] highlights the vital point that the probability that users become victims to a real MITM attack is extremely low.

[1] describes trust relationships and configuration problems in the CA-system. It is a very recent (2013) scan of the entire IPv4 address space, collecting TLS data for analysis. It found 1832 CAs controlled by 683 organizations. Only 7 CAs use name constraints. Three organizations controls 75% of all trusted certificates.

[22] investigated the usage of TLS in smartphone apps and argues that this is an area where client code is easiest to modify and thus clientside pinning (see section 5.2) is feasible. If developers control both ends of a communication there really is no need to rely on the web PKI. Instead pinning of certificates in the app is better.

[23] observes that all proposals to solve the weakest-link problem introduces another authority to check whether the certificate is the correct one. They also note that the insecure status quo can be beneficial for market leaders who are probably not particularly keen on actively helping making themselves obsolete.

Ben Laurie at Google makes a strong case that Certificate Transparency is the best candidate for improving the situation [24]. It is generally applicable, does not push decision onto end user, does not introduce another trusted third party, does not introduce added latency and is migratable.

[25] proposes a new public key validation infrastructure building upon Certificate Transparency and Sovereign Keys.

[26] presents a temporary pinning strategy for TLS until a more permanent solution such as DANE or TACK is in place, which might take years.

[27] is a recent addition to the family of publicly verifiable logs.

[28] highlights the very real possibility that CAs can be legally compelled or coerced into making illegitimate certificates for spy usage. The paper proposes a solution that adopts a trust-on-first-use (TOFU) policy with variations. It also accept certificate changes if the new certificate is issued by same CA or if the CA resides in the same country as previous CA.

The literature review shows that the most popular proposal for improving the authentication component in TLS is DANE because it is mentioned and talked about the most. The main worry about DANE is that it shifts the

required trust from CAs to the registrars, the TLDs and the root: ICANN. One category of proposals is the notaries approach where clients asks so called notaries for history records of seen certificates. Another approach is the pinning strategy whereby clients expects to see certain data or else they alert the user. The third category is public logs where it is expected that entities append their data. Absence of data can be assumed to be an indication of attack in-progress. The rest of the various methods are not meant to replace the CA-system but to co-exist and improve it.

The next chapter will expand on these ideas for alternative authentication methods.

Chapter 5

Alternative authentication methods

What follows are a review of attempts to improve the authentication component in the web PKI. Many of the proposals require an initial secure connection. This is known as a trust-on-first-use (TOFU) leap-of-faith requirement. This requirement works reasonably well in the SSH realm. Not because users verify SSH keys, they do not [29], but because a user interacts only with a small list of servers.

Other limitations on the attempts are Captive Portals and Citibanks. A Captive Portal is a special web page that is shown before using the Internet normally. A typical use of these are wireless hotspots that require payment or user credentials before Internet access.

The Citibank problem is that some sites use many different certificates for a single domain name. The problem got its name because citibank.com first did this and rotated over 100 certificates for unknown reasons.

5.1 DNS-Based Authentication of Named Entities (DANE)

DNS was not built with security in mind, its purpose was simply to map domain names to IP addresses. RFC 3833[1] documents known threats . The

[1] `https://tools.ietf.org/html/rfc3833`

DNS Security Extensions (DNSSEC)[2] is used for securing DNS by adding a digital signature to each DNS resource record (RR) stored in DNS servers. The mapping between domain name and IP address can now be cryptographically secured and you can be reasonably confident you are communicating with the correct IP address after the negotiation is complete.

DNS-Based Authentication of Named Entities (DANE) is an attempt at utilizing the DNSSEC infrastructure to securely transfer public keys for use in TLS [3]. This way, DNS names are bound to public keys, thus bypassing public CAs.

5.2 Pinning

Key pinning is a process where the client in advance knows which key to expect. Typically because a key was pinned on prior visit. More generally it is the act of keeping a history and warn the user if the key changes. Much like how OpenSSH operates. This allows detection of MITM attacks even when CA-signed certificates are used, but only if pinned data is in place. A pin is a relationship between server and a cryptographic identity. Deploying pins carries the risk of accidentally preventing users from reaching the website permanently.

The term "key pinning" is a slight misnomer because the pinned data can be any information. The entire certificate chain or the single public key are candidates here.

Web browsers maintain a preloaded list of pins for high value domains. The Diginotar hack was detected with preloaded pins in the Chrome browser [2].

5.2.1 Trust Assertions for Certificate Keys (TACK)

TACK[4] is a proposed standard for a TLS extension that enables a TLS server to support the "pinning" of a CA key. A client contacting a host will require the server to present a certificate signed by the pinned signing key. TACK requires TOFU and came out in 2012.

[2] `https://tools.ietf.org/html/rfc4033`
[3] `https://tools.ietf.org/html/rfc6698`
[4] `http://tack.io/`

5.2.2 HTTP Public Key Pinning (HPKP)

HPKP[5] is a proposed standard defining a new HTTP header that enables UAs to determine which Subject Public Key Info (SPKI) structures will be present in a web host's certificate chain in future TLS connections.

An example of two pins:

```
Public-Key-Pins:
  pin-sha256="GRAH5Ex+kB4cCQi5gMU82urf+6kEgbVtzfCSkw55AGk=";
  pin-sha256="lERGk61FITjzyKHcJ89xpc6aDwtRkOPAU0jdnUqzW2s=";
  max-age=15768000; includeSubDomains
```

The only allowable cryptographic hash algorithm is sha256. The quoted string is base 64 encoded SPKI Fingerprint. Max-age specifies the number of seconds the UA should regard the host as a known pinned host. The header must be ignored if transferred over non-TLS because otherwise an attack can trivially pin any key. There is a risk that host operator could lose control of their host's private key. In this case the operator would not be able to serve their website in a way that UAs would trust for the duration of the pin's max age. UAs MUST close the connection on pin failure.

In HPKP the pin is in the HTTP headers but it could have been placed anywhere inside the HTTP message. E.g. message size could be reduced if the pins were location at a fixed well-known URL.

5.2.3 Certificate Patrol

Certificate Patrol (CP)[6] is a Firefox addon which implements client controlled pinning. CP pins the entire certificate chain, but can also pin a CA per host. CP is designed to alert users when certificates change or seem suspiciously inconsistent. This solution requires tech-savvy users. The CP website explains:

> Certificate pinning may be considered annoying, but actually it is frequently reminding you that you should be more careful and paranoid.

[5] `https://tools.ietf.org/html/draft-ietf-websec-key-pinning-21`

[6] Certifical Patrol, `http://patrol.psyced.org/`

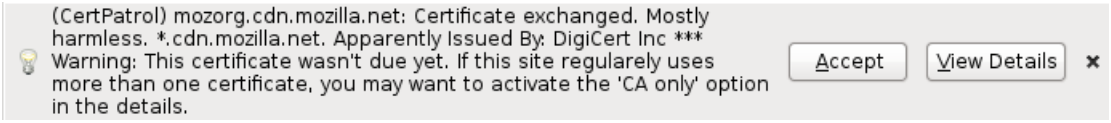

Figure 5.1: Certificate Patrol reports that a new certificate appeared.

CP solves the Citibank Problem (see section 5) by considering it a misconfiguration, and the offending website can be marked as such. These websites can be marked such that any certificate is accepted. After seeing a new and potentially illegitimate certificate, users are presented with a dialog as shown in Figure 5.2. This makes it convenient to inspect never before seen certificates before storing them.

CP is likely to generate too many false positives and become annoying and soon desensitize users.

5.2.4 Certlock

Certlock is a defensive Firefox addon intended to stop a particular attack called *compelled certificate creation attack* [28]. The attacker uses political means to coerce a CA into producing an illegitimate certificate. The intention is surveillance without detection. It functions similar to Certificate Patrol, except it records which country the issuer is from. If the issuing country changes, users are alerted that it can be an attack. The mindbending idea at play here is that some governments may be more trustworthy than others. Recall that trust is the expectancy of a specific behavior. Having trust in a specific government in this regard means expecting that they do *not* coerce CAs to produce illegitimate certificates. Remember this is still a TOFU leap of faith. The geocheck kicks in after a new certificate appears. False negatives occur if attacker coerces actual CA.

The idea can be extended into trusting any geographic region e.g. Europe.

5.2.5 Conspiracy

Kai Engert released Conspiracy[7] , a browser extension in the same spirit as Certlock. [8] The addon shows the country flag of issuers in the certificate

[7] `http://kuix.de/conspiracy/`

[8] `https://addons.mozilla.org/en-US/firefox/addon/107867`

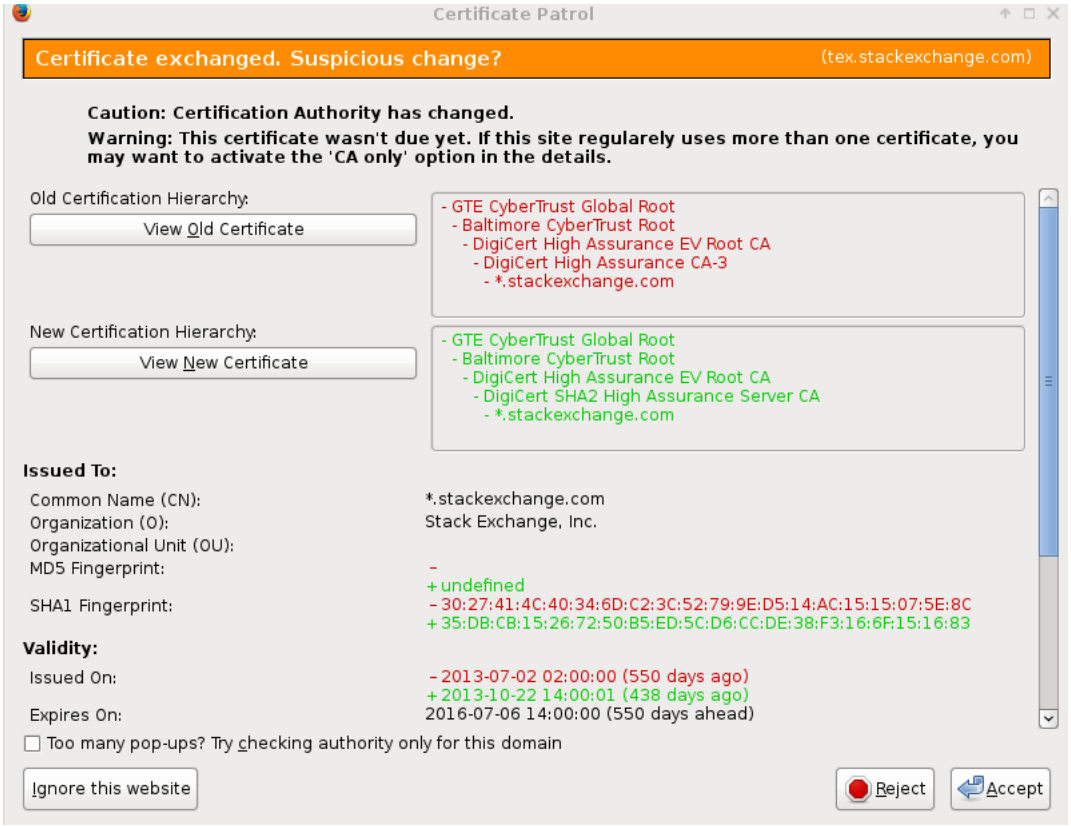

Figure 5.2: Certificate Patrol reports that a new certificate appeared with a different CA.

chain. This helps because a user can decide to not trust CAs from specific countries.

5.3 Notaries

Notaries is a crowd funding type of approach. Nodes in a network exchange data and establishes probable correct views of the world. When clients receive a certificate they can compare it with what the notaries see. It is a consensus rule. It fits the mental models quite well; if everyone around you perceives the same data as you, you are probably not being MITM-ed.

The main advantage that makes these certificate observatories attractive is that website server operators do not have to do anything. This technique

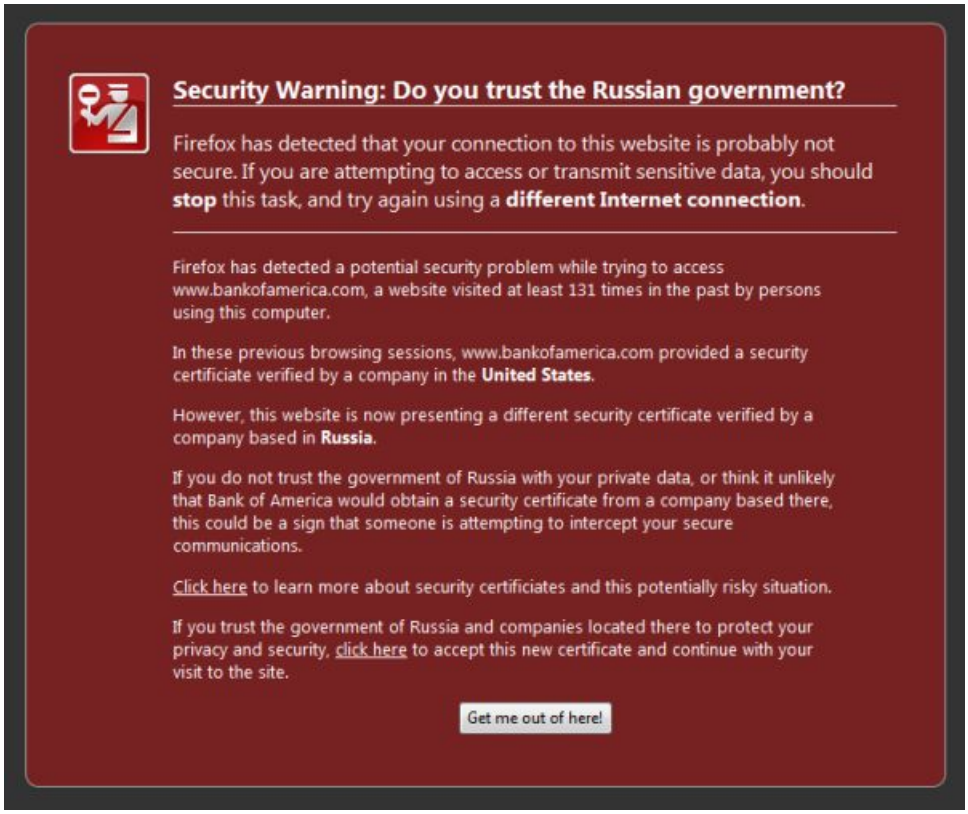

Figure 5.3: Certlock is detecting that the new certificate's issuer is from Russia.

makes it possible to use self-signed certificates without browser warnings.

The main disadvantage is the need for additional connections to query the notaries. Thus latency is increased. This is seen with OCSP where browsers now default soft-fail when they do not get answer from OCSP responder.

Notaries will have problems with Captive Portals and Citibanks.

Chrome has not done these these useless checks by default in recent years [9].

[9] `https://www.imperialviolet.org/2014/04/19/revchecking.html`

Figure 5.4: The Firefox addon Conspiracy showing recent CAs country origin.

5.3.1 Perspectives

In 2008 students from Carnegie Mellon University launched Perspectives. Perspectives establishes a set of public key notaries run by semi-trusted operators [34]. The network notaries periodically probe network services to build a record of the public keys used over time. When a client receives a public key it can contact the notaries and lookup the history of keys used by the service. The data from the notaries are network perspectives that helps users make trust decisions. The primary motivation for creating Perspectives was to help authenticate services that do not have certificates signed by the web PKI.

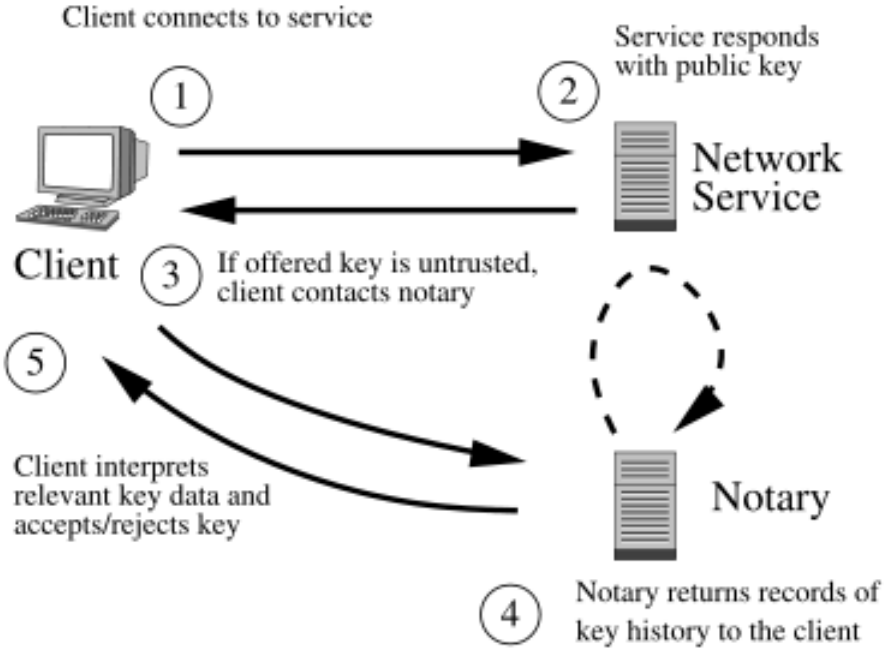

Figure 5.5: Overview of a client using Perspectives. In practice, several notaries would be contacted.

Perspectives pioneered the notary method. Implemented as a Firefox

addon. Per 17. November 2014, Perspectives has around 7000 daily users [10].

5.3.2 Convergence

Figure 5.6: The Convergence logo.

Based on the Perspectives proposal, Moxie Marlinspike implemented a Firefox addon, named Convergence [11] which is a refinement of Perspectives. In Convergence, the certificate that the server provides during the TLS handshake is compared to the values retrieved from multiple notaries. The improvements in contrast to Perspectives are that Convergence is distributed and more anonymous. It is more anonymous in terms of notary operator seeing less of your browsing history. It achieves this by employing a kind of onion routing. A random notary from the list of notaries is selected, then requests are proxied through this notary to other notaries. This is to improve the privacy of the user. Otherwise the notaries can see the entire user browsing history.

The latest commit to the Github repository [12] was on March 7, 2012. The original project is dead, however it is continued in a fork named "Convergence Extra" [13].

5.3.3 Crossbear

Crossbear[14] is a tool to detect and locate TLS MITM attacks [35]. The strategy is to execute network traceroutes from many locations. It is intended as a tool to identify and locate MITM attacks. Its primary purpose is to

[10] `https://groups.google.com/d/msg/perspectives-dev/fLPNSLGAUMc/spNn4_p7LdkJ`

[11] Convergence, `http://www.convergence.io`

[12] `https://github.com/moxie0/Convergence/commits/master`

[13] `https://github.com/mk-fg/convergence`

[14] `https://pki.net.in.tum.de/`

Figure 5.7: The Crossbear logo.

collect data about the MITM attacks, and finally say whether we are dealing with a real problem.

5.4 Certificate log servers

5.4.1 The Sovereign Keys Project

EFF announced the *Sovereign Keys project* in 2011. The proposal extends the current CA system with the possibility of claiming domain names with a *sovereign key*. The keys are recorded in public verifiable logs. A domain name's certificate is only valid if it is signed by the sovereign key.

5.4.2 Certificate Transparency

Certificate Transparency (CT) is a proposal for creating a central audit log of certificates, which is verifiable append-only and maintained by independent monitors [36]. Issuers or users submit new certificates to a public log server. The addition is stored in the form of a signature. Eventually, clients will reject certificates they cannot find in the public log. This requires that it becomes the norm to add certificates to the log and that people monitor it for malicious changes. A domain owner can setup a daily job that inspects the public log for additions of certificates for his domain name, and receive email notifications if fake ones appear. There is going to be large amounts of certificates piling onto the append-only log. For efficient validation a Merkle tree structure is utilized.

5.5 Other approaches

5.6 Mutually Endorsing CA Infrastructure (MECAI)

Kai Engert from Redhat proposed MECAI[15] in 2011 [37]. The primary goal of MECAI is detection of misuse of illegitimate CA-signed certificates. Its strategy is to introduce a second trust opinion. Introduces the concept of shorter lived vouchers issued by existing CAs.

5.6.1 DoubleCheck

DoubleCheck is a Firefox addon and SSH extension which performs a second check of the certificate over the Tor network [33]. If the fetched certificate differs from the first one, an attack in-progress is assumed.

5.6.2 MonkeySphere

Figure 5.8: The monkeysphere logo.

MonkeySphere is a Firefox addon which uses the OpenPGP web-of-trust (WoT) to verify the authenticity of public keys [38]. It naturally enjoys and suffers from the WoT inherent in OpenPGP.

5.6.3 s-links

S-links[16] is a proposal to embed security policy in HTML links [39]. Right now URLs already contain information whether to connect with TLS or not (https:// vs http://). The Google search engine results already pins https:// if available. This idea is extended into HTML attributes:

[15] `https://kuix.de/mecai/`
[16] `http://www.secure-links.org/`

Figure 5.9: The S-links logo.

```
<a link-security="expiry=1357849989;
pin-sha256=YWRmYXNkZmFzZGZhc2RmcXdlcnF3ZXJxd2VycXdlcnF=;
pin-sha256=LPJNul+wow4m6DsqxbninhsWHlwfp0JecwQzYpOLmCQ=;"
href="https://www.example.com">a secure link!</a>
```

Notice that the href is www.example.com delivered over TLS. The embedded pin is the public key of www.example.com and functions as a secure introducer. Additionally s-links enable secure resource loading e.g. external Javascript libraries even in the face of a illegitimate CA-signed certificate. The expiry field is analogous to max-age in HSTS. However, in s-links it is a date because links may be cached. Who will set s-links? Probably search engines and social media sites.

Chapter 6

Conclusions

Alternative methods of providing authenticity in TLS are DANE, notaries, pinning and public append-only logs. DANE's advantage is that we already have a system for mapping names to values and it can be utilized if extended with integrity. Its success depends upon DNSSEC. The deployment of DNSSEC is slow, but is deployed on more and more systems. The disadvantage is that it shifts the required trust from CAs to the registrars, the TLDs and the root: ICANN. It is an improvement because fewer entities are trusted but we still need to trust governmental organizations.

The advantage of Notaries is that website operators do not have to do anything and that self-signed certificates can be used. The disadvantage is the need for network querying because it increases latency and opens up for MITM attackers to drop the query lookups. Notaries will have problems with Captive Portals and Citibanks.

Pinning reduces attack surface because website operators can pin which CA is allowed to issue certificates for their domain name. The disadvantage is that it requires an initial secure connection and that it carries the risk of locking clients out of access your service. TACK requires changes to TLS standards while HPKP simply relies on conforming browsers. Certificate Patrol requires expert users which is only a fraction of the population.

Certlock implements the mindbending idea that some governments may be more trustworthy than others. The difference in trust is subjective to each person and many do not trust governments at all. The main disadvantage is that false negatives occur if attacker coerces the CA and the CA resides in country that the user trusts.

Google's leverage make it likely that Certificate Transparency will take

off. A disadvantage of public logs is that it requires website operators to monitor them, thus placing a great trust in them.

DoubleCheck relies on the Tor network and MonkeySphere relies upon the OpenPGP web of trust. S-links advantage is that it solves the TOFU problem and can introduce stricter security policies.

Chapter 7

Future work

The various pinning strategies seems promising because they add defense in depth. Not all websites require equal amounts of security. Server-controlled pins are a way for high value websites to increase their security. Security-controlled pins combined with variants of s-links is an area worthy of further research.

Bibliography

[1] Zakir Durumeric, James Kasten, Michael Bailey, J. Alex Halderman *Analysis of the HTTPS certificate ecosystem* IMC '13: Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013.

[2] Fox-IT BV. *Black Tulip Report of the investigation into the DigiNotar Certificate Authority breach.* 2012 `http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2012/08/13/black-tulip-update/black-tulip-update.pdf`

[3] Langley, A *Google Online Security Blog - Enhancing Digital Certificate Security* 2013 `http://googleonlinesecurity.blogspot.no/2013/01/enhancing-digital-certificate-security.html`

[4] Mozilla Bug 724929 *Remove Trustwave Certificate(s) from Trusted Root Certificates* 2012. `https://bugzil.la/724929`

[5] Comodo *Comodo Report of Incident on 15-MAR-2011* 2011. `https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html`

[6] B.A. Kitchenham, *Guidelines for performing systematic literature reviews in software engineering* Tech. Rep., EBSE–2007–001, UK, July, 2007.

[7] Diana Berbecaru and Antonio Lioy *On the Robustness of Applications Based on the SSL and TLS Security Protocols*. Springer-Verlag Berlin Heidelberg, 2007.

[8] Peter Gutmann *Engineering Security - Book Draft*, Unpublished, 2014. `https://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf`

[9] Internet Engineering Task Force (IETF) *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. `https://tools.ietf.org/html/rfc6960`

[10] de la Hoz, E.; Cochrane, G.; Moreira-Lemus, J.M.; Paez-Reyes, R.; Marsa-Maestre, I.; Alarcos, B. *Detecting and defeating advanced man-in-the-middle attacks against TLS*. Cyber Conflict (CyCon 2014), 2014 6th International Conference On, 2014.

[11] Ekechukwu, C. ; Lindskog, D. ; Ruhl, R. *A Notary Extension for the Online Certificate Status Protocol*. Social Computing (SocialCom), 2013 International Conference On, 2013.

[12] Oppliger, R. *Certification Authorities Under Attack: A Plea for Certificate Legitimation* . Internet Computing, IEEE Volume: 18 , Issue: 1, 2014.

[13] Braun, J. ; Rynkowski, G. *The potential of an Individualized Set of Trusted CAs: Defending against CA Failures in the Web PKI*. Social Computing (SocialCom), 2013 International Conference on, 2013.

[14] Clark, J. ; van Oorschot, P.C. *SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements*. Security and Privacy (SP), 2013 IEEE Symposium on, 2013.

[15] Puangpronpitag, S. ; Sriwiboon, N. *Simple and Lightweight HTTPS Enforcement to Protect against SSL Striping Attack*. Computational Intelligence, Communication Systems and Networks (CICSyN), 2012 Fourth International Conference on, 2012.

[16] Wazan, A.S. ; Laborde, R. ; Barrere, F. ; Benzekri, A. *The X.509 trust model needs a technical and legal expert*. Communications (ICC), 2012 IEEE International Conference.

[17] Farrell, S. *Not Reinventing PKI until We Have Something Better*. Internet Computing, IEEE, 2011.

[18] Roosa, S.B. ; Schultze, S. *Trust Darknet: Control and Compromise in the Internet's Certificate Authority Model*. Internet Computing, IEEE, 2013.

[19] Internet Engineering Task Force (IETF) *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. `https://tools.ietf.org/html/rfc5280`

[20] Martn Abadi, Andrew Birrell, Ilya Mironov, Ted Wobber, Yinglian Xie *Global authentication in an untrustworthy world*. HotOS'13: Proceedings of the 14th USENIX conference on Hot Topics in Operating Systems,USENIX Association , 2013.

[21] Bernhard Amann, Robin Sommer, Matthias Vallentin, Seth Hall *No attack necessary: the surprising dynamics of SSL trust relationships*. ACSAC '13: Proceedings of the 29th Annual Computer Security Applications Conference, 2013.

[22] Sascha Fahl, Marian Harbach, Henning Perl, Markus Koetter, Matthew Smith *Rethinking SSL development in an appified world*. CCS '13: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013.

[23] Axel Arnbak, Hadi Asghari, Michel Van Eeten, Nico Van Eijk *Security Collapse in the HTTPS Market*. ACM, 2014.

[24] Ben Laurie *Certificate Transparency*. ACM, 2014.

[25] Tiffany Hyun-Jin Kim, Lin-Shung Huang, Adrian Perring, Collin Jackson, Virgil Gligor *Accountable key infrastructure (AKI): a proposal for a public-key validation infrastructure*. International World Wide Web Conferences Steering Committee, 2013.

[26] Gabor X Toth, Tjebbe Vlieg *Public Key Pinning for TLS Using a Trust on First Use Model*. University of Amsterdam, 2013. `http://145.100.102.229/rp/2012-2013/p56/report.pdf`

[27] P Szalachowski, S Matsumoto, A Perrig *Public Key Pinning for TLS Using a Trust on First Use Model*. Proceedings of the 2014 ACM.

[28] C Soghoian, S Stamm *Certified Lies: Detecting and Defeating Government Interception Attacks*. Financial Cryptography and Data Security, Springer, 2012.

[29] Peter Gutmann *Do Users Verify SSH Keys?* Department of Computer Science at the University of Auckland. 2011. `https://www.usenix.org/system/files/login/articles/105484-Gutmann.pdf`

[30] Marlinspike, M *More Tricks for Defeating SSL in Practice*. Black Hat USA Talk, 2009.

[31] Internet Engineering Task Force (IETF) *HTTP Strict Transport Security (HSTS)*. https://tools.ietf.org/html/rfc6797

[32] Electronic Frontier Foundation *HTTPS Everywhere*. https://www.eff.org/https-everywhere

[33] Mansoor Alicherry Angelos D. Keromytis, *DoubleCheck: Multi-path Verification Against Man-in-the-Middle Attacks*, Department of Computer Science, Columbia University in the City of New York, 2009. http://www1.cs.columbia.edu/~angelos/Papers/2009/doublecheck.pdf

[34] D. Wendlandt, D. Andersen, A. Perrig. *Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing*. USENIX Annual Technical Conference, 2008. https://www.cs.cmu.edu/~perspectives/perspectives_usenix08.pdf

[35] Ralph Holz, Thomas Riedmaier, Nils Kammenhuber, Georg Carle *X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-Middle*. ESORICS ,(2012).

[36] Langley, A *ImperialViolet - Certificate Transparency*. 2011. https://www.imperialviolet.org/2011/11/29/certtransparency.html

[37] Kai Engert, *MECAI - Mutually Endorsing CA Infrastructure*, Redhat, 2011. http://kuix.de/mecai/mecai-proposal-v2.pdf

[38] *The Monkeysphere Project* http://web.monkeysphere.info/

[39] Joseph Bonneau, *S-links: Why distributed security policy requires secure introduction* Google Inc, 2013. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41138.pdf